Mobility Prediction-based Smartphone Energy Optimization for Everyday Location Monitoring

Yohan Chon yohan@cs.yonsei.ac.kr
Elmurod Talipov elmurod@cs.yonsei.ac.kr
Hyojeong Shin hjshin@cs.yonsei.ac.kr
Hojung Cha hjcha@cs.yonsei.ac.kr

Department of Computer Science
Yonsei University
Seoul, Korea

Abstract

Monitoring a user's mobility during daily life is an essential requirement in providing advanced mobile services. While extensive attempts have been made to monitor user mobility, previous work has rarely addressed issues with battery lifetime in real deployment. In this paper, we introduce SmartDC, a mobility prediction-based adaptive duty cycling scheme to provide contextual information about a user's mobility: time-resolved places and paths. Unlike previous approaches that focused on minimizing energy consumption for tracking raw coordinates, we propose efficient techniques to maximize the accuracy of monitoring meaningful places with a given energy constraint. SmartDC comprises unsupervised mobility learner, mobility predictor, and Markov decision process-based adaptive duty cycling. SmartDC estimates the regularity of individual mobility and predicts residence time at places to determine efficient sensing schedules. Our experiment results show that SmartDC consumes 81% less energy than the periodic sensing schemes, and 87% less energy than a scheme employing context-aware sensing, yet it still correctly monitors 80% of a user's location changes within a 160-second delay.

Categories and Subject Descriptors

C.3 [**Special-purpose and Application-based Systems**]: Real-time and embedded systems

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Mobility Learning, Mobility Prediction, Adaptive Sensing, Energy-Efficient

1 Introduction

Understanding and predicting human mobility in daily life is a fundamental resource for broad-domain applications. For instance, service providers predict a user's behavior (e.g., having lunch or going home) and provide appropriate services in a timely manner (e.g., sending lunch coupons or home preheating). Understanding human mobility, in general, provides useful information for traffic engineering, urban planning, predicting the spread of human and electronic viruses, and resource management in mobile communications [9, 24, 25]. All these applications benefit from understanding and predicting human mobility: time-resolved places and paths.

Mobile phones are widely used for tracing human mobility since mobile phones (1) have almost 100% penetration, (2) are closely tied to daily life, and (3) are capable of locating themselves using various approaches. The Global Positioning System (GPS) and Wireless Positioning System (WPS) using cell tower and Wi-Fi access points (AP) are common technologies that provide a user's raw coordinates (i.e., latitude and longitude) [1, 14]. Ambient fingerprints are often constructed to recognize semantic places with room-level accuracy using radio beacons (e.g., cell towers, Wi-Fi APs, and Bluetooth) and surrounding factors (e.g., light, color, texture, and sound patterns) [2, 12]. A simple choice for monitoring mobility is to periodically sense a user's location context. Such a scheme, however, significantly reduces the battery's lifetime in mobile devices. To optimize energy consumption for continuous sensing, various approaches have been proposed. These include sensor selection by movement detector using accelerometers [12, 18, 21, 28], minimizing energy consumption within accuracy requirements [17, 21], minimizing location error for a given energy budget [7, 18, 28], and utilizing a prediction-based approach [7, 17, 19]. While extensive attempts have been made to continuously examine a user's mobility with less energy consumption, we argue that previous work did not fully consider regular patterns in human mobility to reduce energy consumption in real deployments.

Our research goal is to develop a framework that continuously provides location context with minimum energy consumption. The key motivations of our work are as fol-

Table 1. Summary of Features in Related Systems for Everyday Location Monitoring

System	Major Context	Sensor Selection						Adaptive Scheduling Policy (Constraints)	Mobility Prediction	Evaluation Platform
		Movement Detector		*Path Tracking*		*Place Recognition*				
EnLoc [7]	Raw Coordinates	-	-	GSM GPS Wi-Fi	Adaptive	-	-	Minimize Location Error (Energy Budget)	Heuristic	Unknown
Jigsaw [18]	Raw Coordinates	Accel[a]	Always	GPS	Adaptive	-	-	Minimize Location Error (Energy Budget)	-	Apple iPhone Nokia N95
Zhuang [28]	Raw Coordinates	Accel	Always	GPS WPS[b]	Adaptive	-	-	Adjust GPS Sample Rate (Battery Level)	-	HTC G1
RAPS [21]	Raw Coordinates	Accel	12.5%	GPS	Adaptive	-	-	Minimize Energy Consumption (Location Error)	-	Nokia N95
a-Loc [17]	Raw Coordinates	-	-	GPS WPS	Adaptive	-	-	Minimize Energy Consumption (Location Error)	HMM	HTC G1
iLoc [19]	Meaningful Place	GSM Wi-Fi	30sec 2min	GPS	Always	GSM Wi-Fi	30sec 2min	Fixed Interval	Viterbi Algorithm	P4-2GHz PC
SensLoc [12]	Meaningful Place	Accel	50%	GPS	10sec	Wi-Fi	10sec	Fixed Interval	-	HTC G1
SmartDC	Meaningful Place	GSM Wi-Fi	2min Adaptive	GPS WPS	Adaptive	Wi-Fi	Adaptive	Maximize Place Detection (Energy Budget)	MDP	HTC Desire

a. Accelerometer, b. Wireless positioning systems provided by Android platform

lows: (1) finding meaningful places is a key target of human-centric mobile services, since people spend approximately 87% of their time indoors [13]; and (2) human mobility is amenable to prediction because of habitual time-resolved movements with reasonably small variation [9]. Thus, we focus on monitoring meaningful places (i.e., points of interest, or POIs) using the regularity of individual mobility pattern. The main idea is that the system senses location context based on a predicted schedule; that is, when the movement to the next location will take place. In other words, when a user visits a place our system makes predictions on departure times (i.e., residence time in the place). The key technical challenges are (1) simultaneous learning and predicting a user's mobility, (2) predicting human mobility with low complexity, and (3) minimizing energy consumption.

In this context, this paper proposes SmartDC: mobility prediction-based adaptive duty cycling for everyday location monitoring. SmartDC comprises three components: *mobility learner*, *mobility predictor*, and *adaptive duty cycling*. *Mobility learner* uses unsupervised learning to incrementally collect mobility patterns in colloquial terms. Based on our previous work [5, 6], we developed a personalized scheme that collects POI's raw coordinates and also recognizes POIs with room-level accuracy. *Mobility predictor* uses a location predictor to predict departure time to the next location. We implemented several location predictors and compared their cost and performance. *Adaptive duty cycling* uses a Markov decision process (MDP) to determine the efficient sensing moment for a given energy budget. The proposed scheme maximizes the accuracy of mobility monitoring based on the regularity in individual mobility. The primary contributions of our work are:

- Designing simultaneous and incremental mobility learning and prediction components (Section 3).
- An extensive performance analysis of several location predictors for the estimation of predictable regularity in human mobility (Section 4).
- Implementing a real system that maximizes the robustness, accuracy, and scalability of everyday location monitoring for a given energy budget (Section 4).

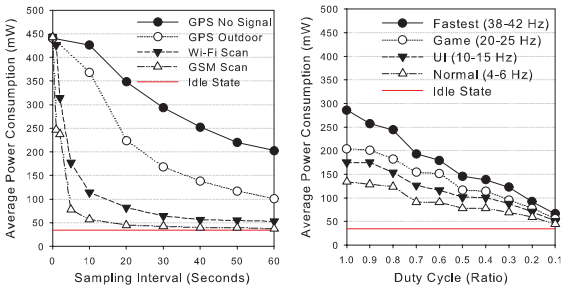
We implemented SmartDC on the Android framework as a service, and evaluated the system using traces of over four weeks for 8 participants in real environments. While the performance depended on a given energy budget, our extensive evaluation showed that SmartDC simultaneously performed mobility learning and prediction, and outperformed previous systems in terms of accuracy versus energy consumption.

2 Preliminaries and Related Work

Several research studies found in the literature considered energy-efficient sensing for tracing a user's everyday location. We first assess related works, and then elaborate on our motivation with highlights from previous work on the issues at hand.

Table 1 summarizes the key features of related systems. According to the target context, we categorized context-monitoring systems into two groups: raw coordinates and meaningful places. To reduce energy consumption, most systems use a movement detector to activate different sensors in both moving and stationary states [12, 18, 19, 21, 28]. The user's speed estimated by GPS is also commonly used to switch around localization techniques [12, 21]. Based on detected movement, most systems use a power-intensive sensor, such as a GPS, if the user is moving or if the estimated location includes a considerable amount of uncertainty.

To track a user's raw coordinates, many approaches have been proposed to minimize energy consumption with adaptive sensor scheduling [7, 17, 18, 21, 28]. EnLoc [7] uses a dynamic programming technique to minimize location error for a given energy budget. A heuristic approach with a mobility tree is used to predict the next sensing time. The system, however, does not implement incremental learning, but instead uses a manually constructed mobility tree. Jigsaw [18] uses the adaption technique to adjust the GPS sampling rate (i.e., 20, 10, 5, 2, 1 mins., and 5 secs.). The system recognizes a user's activity using an accelerometer signal and uses MDP process to minimize location error within a given energy budget. Zhuang *et al.* [28] use location profiles and battery levels to adjust the GPS sampling rate. With these schemes, however, none uses an unsupervised prediction module or recognizes a place with room-level accuracy. Works in [17, 21] minimize energy consumption by adaptively using a GPS, only if the estimated location error ex-



(a) (b)

Figure 1. HTC Desire energy profile. We measured power consumption of several sensors, such as (a) GSM, Wi-Fi, and GPS with periodic sampling; and (b) accelerometer with duty cycle. The profile data and test software are downloadable from our website (http://lifemap.yonsei.ac.kr).

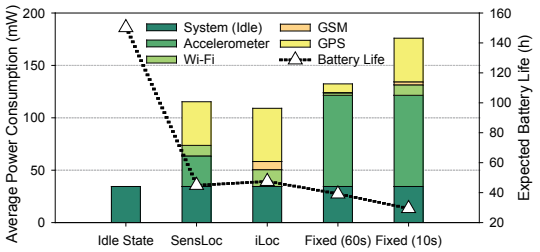


Figure 2. Power consumption analysis on work related to mobility learning. We used proposed parameters of related works as shown in Table 1. The fixed scheme uses a fixed time interval. We assumed that the GPS is turned on only if a user is moving.

ceeds an accuracy threshold. To estimate uncertainty in a location, each system uses different approaches. A-Loc [17] uses the hidden Markov model (HMM) to predict user movement with an accuracy model. RAPS [21] uses moving distance, space-time history, and cell tower-based blacklisting. In contrast to these systems, SmartDC uses an individual mobility pattern to estimate the uncertainty of a user's behavior without powering up the sensors. The proposed system optimizes a sensing schedule to maximize the accuracy of mobility monitoring within a given energy constraint. Additionally, we focus on detecting meaningful places, which is an essential component in advanced mobile services.

To recognize meaningful places, ambient fingerprints have been proposed to provide POIs with room-level accuracy. SensLoc [12] uses the Tanimoto coefficient of the Wi-Fi vector to detect the entrance and departure time. The system reduces energy consumption based on sensor selection, but the scheme does not use mobility prediction to optimize the sensing schedule. iLoc [19] uses the cosine similarity of GSM and Wi-Fi vectors to detect movement and meaningful places. The system uses the Viterbi algorithm to predict future location, but sensor scheduling is not considered in reducing the energy consumption. We analyzed these approaches to design a mobility learner, and to develop an adaptive sensor scheduling based on mobility prediction.

To the best of our knowledge, SmartDC is the first system that applies a simultaneous mobility learning and prediction scheme to mobile phones. The proposed system gradually learns a user's mobility pattern, and optimizes sensing schedule using the predictable regularity of individual behavior.

2.1 Limitation of Mobility Learning

Continuous mobility learning without duty cycling is not a viable scheme because of the significant energy usage. We first measured the average power consumption of each sensor to estimate the energy cost of mobility learning. To maintain the measurement consistency, we used the Monsoon Power Monitor to deliver a constant voltage of 3.7 V and halted all applications except for a system service. Figure 1 shows the energy profile of HTC Desire. The idle state consumes 34.5 mW. Sensors are listed in the order of energy consumption: GPS, accelerometer, Wi-Fi, and GSM. The mobile phone performs GSM scans without additional energy cost since it always maintains a cell tower connection for voice communications. The duty cycle of the accelerometer should be carefully designed, since operating the accelerometer can incur significant energy usage. The application should lock the CPU to continuously use the accelerometer and prevent sleep state for continuous sensing. Although the accelerometer itself uses very little power, the continuous use of accelerometer needs to keep the CPU as well as associated high-power components active to access the sensor data [22]. Thus, computing the accelerometer signal with a 50% duty cycle at the lowest sampling rate (4-6 Hz) for 10 minutes consumes more energy than turning on a GPS with 1-minute intervals for 5 minutes. For this reason, most of the mobile platforms (e.g., Android, iPhone, and Nokia Maemo) restrict continuous sampling of acceleration while the screen is turned off [15]. Although a dedicated microcontroller may reduce sensing energy [22], even the latest smartphones do not employ such additional processor for sensing purpose. The proposed scheme, therefore, does not use an accelerometer for the pragmatic reason.

Based on the energy profile, we estimated the energy consumption of various mobility learning schemes. We made two assumptions: (1) A user spends 3 hours to move around outdoors each day [13]; and (2) the movement detector always recognizes user movements correctly. For example, SensLoc uses GPS and Wi-Fi every 10 seconds while a user moves, and the system activates an accelerometer with a 50% duty cycle when a user is stationary. Thus, the stationary state consumes 67.8 mW, and the moving state consumes 447.8 mW, which is derived from the sum of the idle state (34.5 mW), the GPS reading every 10 seconds outdoors (333.2 mW), and the Wi-Fi scanning every 10 seconds (80.1 mW). The average power consumption is 447.8 mW$\times\frac{3}{24}$+67.8 mW$\times\frac{21}{24}$=115.3 mW.

Figure 2 presents the power consumption of several schemes. The expected battery lifetime is 29 to 48 hours if a smartphone runs only mobility learning schemes. A previous study showed that, without mobility learning, 60% of people used their smartphone from 14 to 41 hours with a single battery charge [8]. This means that mobility learning may reduce battery lifetime by at least 16%, and by 53% at most. Such energy consumption is a burden to users, since mobility learning is not a primary function of smartphones. Consider-

ing that the expected lifetime of idle state is 150 hours, previous learning schemes have room for lifetime improvement. The optimal scenario is that the system turns on sensors only when a user changes his/her states (i.e., entrance and departure moments). Our main idea is to adaptively sense the moment that includes a considerable possibility of state change. We predict the state change using the regularity of individual mobility patterns. The proposed system uses an energy budget as a constraint in order to customize sensing schedules to diverse smartphone usage.

2.2 Selection of Mobility Predictor

The ability to predict human mobility is critical for various mobile computing, such as routing in ad hoc wireless networks [27], resource provisioning in wireless networks [23, 16], and predicting the spread of biological pathogens or electronic viruses [26].

Extensive researches have been made on mobility prediction, especially in the networking community. Previous schemes for mobility prediction can be classified into three groups: Markov-based schemes [17, 19, 20], compression-based schemes [29], and neural networks-based schemes [4]. Works in [25] evaluated the performance of various mobility predictors using a two-year trace of over 6,000 users on Dartmouth's campus-wide Wi-Fi wireless network. They found that the complex predictors were at best only negligibly better than the simple Markov predictor in practice. Indeed, a second-order Markov model, meaning that a predictor attempts to estimate the next location from two of the most recent locations, was the most accurate technique of all the examined techniques. In addition, the simple predictor with the Markov-based scheme is appropriate for use on resource-constrained devices, such as smartphones, because of low computation overheads and modest storage requirements [20]. Considering that our goal is to minimize energy consumption on mobile devices, we implemented and compared simple predictors that included a second-order Markov predictor.

3 SmartDC System

We first describe SmartDC usage scenario, then present the technical details. SmartDC runs as a background service in mobile phones. The energy budget for SmartDC is initially defined as E (i.e., percentage of remaining battery charge), but a user may manually change it. When a user stays at a place for a certain period of time, SmartDC considers it a meaningful place and generates a node with place signatures that include location, Internet connectivity, visiting time, residence-time, and the Wi-Fi fingerprints. The system incrementally constructs users' meaningful places with room-level accuracy in their daily lives, and recognizes the revisited places using the stored place signatures. With SmartDC, the mobile phone can always notify/share the change of places to internal and third-party applications.

Figure 3 shows the overall architecture of SmartDC. The system comprises mobility learner, mobility predictor, and adaptive duty cycling. In the mobility learner, we use GSM, Wi-Fi, and GPS which are the common components in current mobile phones. We define three levels of sensing according to the sensor's energy consumption. The mobility

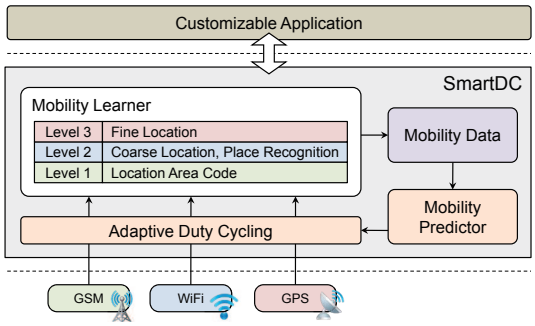


Figure 3. System architecture of SmartDC.

learner activates high-level sensing only if low-level sensing fails to obtain location information. In the context of this work, location is a room-level place. We use the similarity of Wi-Fi fingerprints to classify places as room-level. Initially, SmartDC uses a fixed time interval (e.g., 2 minutes) to collect a user's mobility data. Based on the collected mobility data, the mobility predictor predicts residence time at locations and rewards for each location sensing. Finally, adaptive duty cycling makes use of a dynamic programming technique to determine a sensing schedule and to maximize sensing rewards for a given energy budget. With the predicted sensing schedule, the system monitors the exceptions to the predicted patterns. When an exception occurs the system switches to a sensing mode with a fixed interval for mobility learning. Consequently, SmartDC simultaneously performs learning and prediction to minimize energy consumption with robust location monitoring.

3.1 Problem Definition

The key problem in location monitoring is choosing an optimal location-sensing interval. We formulated the location-sensing policy as a Markov decision process (MDP). An MDP is a stochastic process that contains a 4-tuple $\langle \mathbb{S}, \mathbb{A}, P, \mathbb{R} \rangle$. The finite set of states in which $\mathbb{S} = \{l_t, l_{t+1}, \cdots\}$ is a set of meaningful places that were previously visited by the user. Action set $\mathbb{A} = \{a_t, a_{t+1}, \cdots\}$ is a set of actions taken on states, i.e., the length of time before the next location sensing. $P_a(l_t, l_{t+1})$ is the transition probability from state l_t to l_{t+1}, when action a is taken. $\mathbb{R} = \{R_{a_t}(l_t, l_{t+1}), R_{a_{t+1}}(l_{t+1}, l_{t+2}), \cdots\}$ is an expected reward at the transition between the states for the action taken. The reward is location-monitoring accuracy, and our goal is to maximize the cumulative function of rewards, defined as follows:

$$\text{maximize } \sum_{t=0}^{\infty} R_{a_t}(l_t, l_{t+1}) \text{ subject to } C \leq E,$$

where C is the total consumed energy for the location monitoring process and E is a given energy constraint. The solution for this problem involves designing an optimal policy π that will be explained in Section 3.4. Figure 4 illustrates the defined problem.

3.2 Mobility Learner

The role of mobility learner is to collect individual mobility history without impending users. Although the concept of mobility learner is similar to the previous work [12, 19], we designed sensing levels to minimize the usage of power-intensive sensors. The basic idea is that high-level sensing

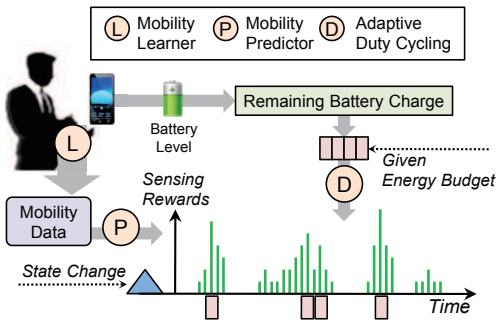


Figure 4. The conceptual problem. The sensing rewards are derived using residence-time and sensing cost from the mobility history. The goal is to allocate energy budget to maximize the accuracy of state change.

is activated only if low-level sensing fails to obtain accurate information.

The first level uses GSM to obtain the Location Area Code (LAC) [1] to detect exceptions within the predicted sensing schedule. The first level continuously monitors the LAC with minor energy consumption, since a mobile phone basically updates the LAC for voice communication. The system does not activate the second level until the next sensing time, if the observed LAC follows a predicted sequence-pattern. Otherwise, if the first level detects an exception, the system immediately uses the second level to collect a new pattern of individual mobility. For example, if a user normally goes to the office on weekday mornings, SmartDC only turns on Wi-Fi near the entrance of the office if the expected LAC is observed. If a user goes on a business trip, the system uses the mobility learner when it detects a new pattern of LAC.

The second level uses Wi-Fi scanning to recognize a change of places and revisited places. The basic operation is that if a user is stationary, the signal fingerprints of surrounding Wi-Fi APs are relatively similar to each other. We use a scan window to perform multiple scans to tolerate noisy signals [11, 12, 19]. Given window size w and sampling interval t_w, the Wi-Fi similarity function $\mathcal{S}$ is defined using the Tanimoto Coefficient, which is a widely used technique for measuring the similarity between two vectors [10, 12]:

$$\mathcal{S} = \begin{cases} \text{different (move)} & \text{,if} \frac{\vec{f}_{t_1} \cdot \vec{f}_{t_2}}{\|\vec{f}_{t_1}\|^2 + \|\vec{f}_{t_2}\|^2 - \vec{f}_{t_1} \cdot \vec{f}_{t_2}} \leq \varphi \\ \text{same (stationary)} & \text{,else} \end{cases}$$

where $\vec{f}_{t_i}$ is the Wi-Fi vector scanned at t_i for duration w, φ is the similarity threshold, and the output is a similarity estimated between 0.0 to 1.0. The movement detector uses $\mathcal{S}(\vec{f}_{t-1}, \vec{f}_t)$ to detect change of places, while place recognition uses $\mathcal{S}(\vec{f}_{l_i}, \vec{f}_{l_j})$ to identify a visited place where $\vec{f}_{l_i}$ and $\vec{f}_{l_j}$ is the Wi-Fi vector scanned at place l_i and l_j respectively. The system generates meaningful places when it detects the stationary state. When a user revisits a place, the system aggregates mobility data and reuses physical location information without activating additional sensors. The second level also uses wireless communication to obtain coarse location from the WPS provided by Android.

The third level activates GPS to acquire fine location, if the system fails to get accurate location in the second level. Algorithm 1 presents the pseudo-code of mobility learner.

[1] LAC is a unique number within a cellular radio network. Location areas comprise one or several radio cells.

Algorithm 1 Three level sensing in mobility learner.

Input: Previous fingerprint $\vec{f}_{t-1}$, predicted pattern $\mathbb{P}$
Output: Current location l_t
Notation: LAC ℓ_t, fingerprint $\vec{f}_t$, movement state s_t, similarity threshold φ, location-accuracy threshold ϕ

```
ℓ_t ← getLAC()                                   # Level 1 using GSM
if ℙ.contains(ℓ_t) then
    return
end if
f⃗_t ← scanWiFi()                                 # Level 2 using Wi-Fi
if 𝒮(f⃗_{t−1}, f⃗_t) ≤ φ then
    s_t ← Move
else
    s_t ← Stationary
end if
l_t ← getCoarseLoction()                         # Level 2 using WPS
l_t′ ← makeLocation(s_t, l_t, f⃗_t)               # Place Recognition
if l_t′.getAccuracy() ≤ ϕ then
    return l_t′
end if
l_t′ ← getFineLocation()                         # Level 3 using GPS
return l_t′
```

3.3 Mobility Predictor

The role of mobility predictor is to estimate the action set $\mathbb{A}$ and transition probabilities P to each possible next location. In other words, when a user visits a place, the system makes predictions about departure times (i.e., residence-time in a place). Our mobility predictor outputs a list of place and probability pairs to determine an efficient sensing schedule, instead of a future location with the maximum probability.

We use two families of location predictor: Order-k Markov, and LZ-based [29]. In predictions for human mobility, Song *et al.* [25] found that simple predictors were better than complex predictors in practice. Thus, we chose $O(1)$ Markov, $O(2)$ Markov, and LZ-based predictors, which were efficient in accuracy and memory usage aspects. Our scheme contributes to the prediction of human movements with room-level accuracy, while previous works [9, 24, 25] used associated cell-tower or AP for a user's location.

We briefly describe the order-k Markov and LZ-based predictors. The order-k Markov predictor estimates the next location from a current context, which comprises the k most recent locations. Accordingly, given current location context $l_{t-k+1} \cdots l_{t-1} l_t$, the transition probability to a future location l_{t+1} is expressed as:

$$P(l_t, l_{t+1}) = P(X_{t+1} = l_{t+1} | X_t = l_{t-k+1} \cdots l_{t-1} l_t) =$$
$$P(X_{i+k+1} = l_{i+k+1} | X_t = l_{i+1} \cdots l_{i+k-1} l_{i+k}) = \frac{N(X_t l_{t+1}, L)}{N(X_t, L)},$$

where $N(s, L)$ denotes the number of times the substring s occurs in the context L. The formulation indicates that the probability is the same anywhere the context of k recent locations is the same.

The LZ-based predictor is based on an incremental parsing algorithm by Ziv and Lempel [29]. The LZ-based predictor is similar to the order-k Markov predictor, except that k is a variable allowed to grow to infinity [3]. Let γ be the

empty string and L the input movement history. The LZ parsing algorithm takes L and splits it into substrings $s_0 s_1 \cdots s_m$ such that $s_0 = \gamma$, and for all $j \geq 1$ the prefix of substring s_j without its last symbol is equal to s_i, for all $i < j$. Note that the partitioning is made sequentially, so when each s_i is determined the algorithm then considers the remaining trace. The LZ algorithm builds a tree, called an LZ tree, to store these substrings (i.e., mobility patterns). Each tree node represents a substring and stores the number of times that the substring appears among the patterns parsed by the LZ algorithm. Then, given current location context l_t the probability to next location l_{t+1} is expressed as:

$$P(l_t, l_{t+1}) = \frac{N^{LZ}(l_t l_{t+1}, L)}{N^{LZ}(l_t, L)},$$

where $N^{LZ}(s, L)$ denotes the frequency of the substring s in the LZ tree.

Instead of choosing the location with the highest probability, we extract all locations that have the transit probability (from the location at t) in order to determine an efficient sensing schedule. Let n be the number of extracted locations, we extract the action set $\mathbb{A} = \{a_0, \cdots, a_n\}$ that is a previously-observed residence time at current location l_t. Based on estimated transition probability and the action set, mobility predictor generates a *waiting-time distribution* $P_{l_t}(\Delta t)$, where Δt is the time spent by a user at the current location. To cover the error from the mobility learning, we add a Gaussian noise to the generated $P_{l_t}(\Delta t)$. The noise distribution is empirically found in real-trace as described in Section 4.3. Then, discrete action set in one-minute interval is defined as $\mathbb{A}' = \{a_i | 0 \leq i \leq n, P_{l_t}(a_i) > 0\}$, where n is the number of quantized actions that have positive probability.

3.4 Adaptive Duty Cycling

The goal of adaptive duty cycling is to maximize the accuracy of monitoring mobility, and to optimize a sensing policy on diverse smartphone usage with a given energy budget.

Randomness is inherent in human movements although humans tend to move with reasonably small variation. In other words, a user does not always follow the previously-observed movement patterns. Thus, the accuracy of the prediction-based adaptive duty cycling may decrease due to the randomness of human behavior. To overcome this problem, we split a given energy budget E into two parts: energy for prediction E_{PRD} (i.e., the case of following patterns in mobility history), and energy for exception E_{EXP} (i.e., the case of moving with a new pattern). To split E, we use potential predictability in individual mobility. We measure predictability by comparing the current pattern to the historical data. For example, potential predictability is 0.7 if a user follows the previously observed pattern 7 times out of a total of 10 visit counts. Then, the exception ratio is 0.3 and the system allocates 30% of given energy for exception monitoring.

We first present the sensing policy using E_{PRD}. Let $V(l_t, E)$ be an optimal sensing schedule at state (l_t, E), which is also the maximal sensing rewards at the current location l_t within a given energy budget E. When the system detects entrance at a certain place l_t, the mobility predictor estimates the waiting-time distribution $P_{l_t}(\Delta t)$ that is used for reward function $R_{\mathbb{A}'}(l_t)$. The sensing cost c_i is an estimated energy consumption at a_i, depending on the stored accuracy of location information. If the accuracy is worse than location-accuracy threshold ϕ, the cost is the sum of sensing levels 2 and 3. Otherwise, the cost is the energy consumption of level 2. Our goal is to maximize the cumulative function of rewards, defined as follows:

$$\text{maximize } \textstyle\sum_{i=0}^{|\mathbb{A}'|} R_{a_i}(l_t) \text{ subject to } C_{PRD} \leq E_{PRD},$$

where C_{PRD} is the total consumed energy for the location monitoring process and E_{PRD} is a given energy constraint. The solution for this problem involves designing an optimal policy π and computing the value function $V(\cdot)$, which is expressed as:

$$\pi = \arg\max_a \{\textstyle\sum_{i=0}^{|\mathbb{A}'|} P_{a_i}(l_t)(R_{a_i}(l_t) + V_{a_{i+1}}(l_t, e - c_i))\},$$

$$V(l_t, E_{PRD}) = \textstyle\sum_{i=0}^{|\mathbb{A}'|} P_{\pi}(l_t)(R_{a_i}(l_t) + V_{a_{i+1}}(l_t, e - c_i)),$$

where e is the remaining energy budget. In words, optimal sensing moments are allocated to maximize the rewards of sensing within a given energy budget. Algorithm 2 presents the pseudo-code of adaptive duty cycling using the dynamic programming technique. The output of dynamic programming is the set of sensing moment, and inputs are the current time, energy budget, and generated reward function.

Algorithm 2 Allocation policy of energy for prediction

Input: Sensing moment i, remaining energy budget e
Output: Set of sensing moments derived from $V(\cdot)$

```
if e < 0 then                                   # Boundary condition 1
    return invalid
end if
if i > |A'| then                                # Boundary condition 2
    return 0
end if
if V_{a_i}(l_t, e) is valid then                # Memoization
    return V_{a_i}(l_t, e)
end if
V_{a_i}(l_t, e) ← invalid
for k is i + 1 to |A'| do
    V_{a_i}(l_t, e) = max(V_{a_i}(l_t, e), R_{a_k}(l_t) + V_{a_{k+1}}(l_t, e − c_k))
end for
return V_{a_i}(l_t, e)
```

The system allocates E_{EXP} to minimize the longest blank-term (i.e., continuous duration without location sensing). Let $B = \{b_0, b_1, \cdots, b_k\}$ be a set of blank terms, and b_i the length of the i_{th} blank-term in minutes. Given j location sensing opportunities, we allocate it to B to minimize the longest blank-term. For example, we assume that b_i is an 8-length blank-term (oooooooo). Then it can be divided into three shorter blank-terms (b_i, b_{i+1}, b_{i+2}) if two location sensing is allocated to b_i (ooxooxoo). The goal is to minimize the longest length of blank-term, defined as:

$$\text{minimize } \max_{0 \leq i \leq k} b_i \text{ subject to } C_{EXP} \leq E_{EXP},$$

where C_{EXP} is the total consumed energy for dividing all blank terms. Let r be a maximum length in B, we find the optimal r by using a binary search between 0 to $\max_{0 \leq i \leq k} b_i$. When r is fixed, the required energy C_{EXP} for reducing all elements in B to less than r can be calculated in $O(k)$. Then, we decrease r if C_{EXP} is smaller than E_{EXP}, otherwise we increase r. Algorithm 3 presents the pseudo-code of the allocation policy.

Figure 5 illustrates the overall process of prediction-based adaptive duty cycling.

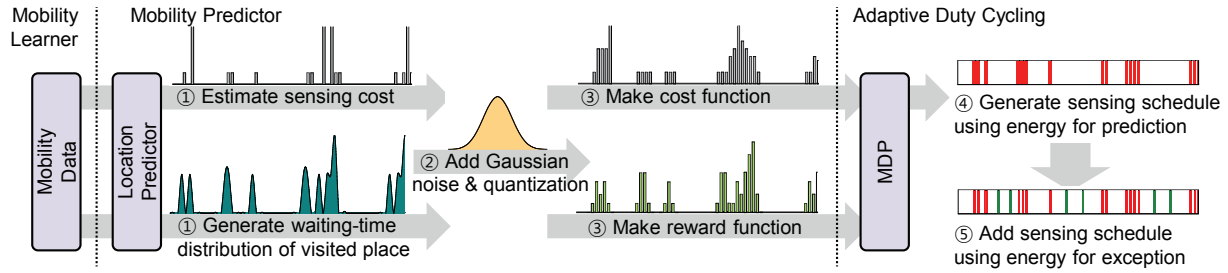
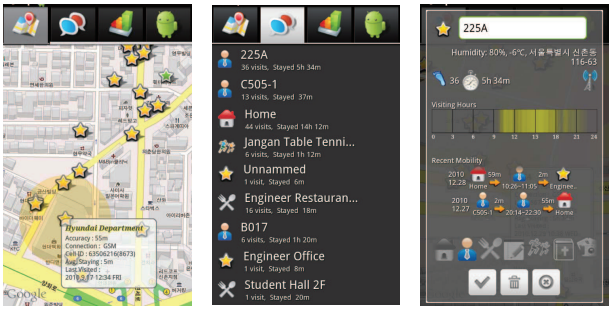


Figure 5. The process of prediction-based adaptive duty cycling. (1) Location predictor generates $P(\Delta t)$ and sensing cost using collected data. (2) Gaussian noise is derived from the empirical error of mobility learner. (3) The system makes quantized rewards and cost functions in minutes. (4) MDP outputs predicted sensing schedule. The system gradually measures an exception ratio to split energy budget into for prediction and for exception. (5) Energy for exception is used to minimize longest blank-term.

Algorithm 3 Allocation policy of energy for exception

Input: Set of blank-term B, energy budget for exception E_{EXP}
Output: Optimal length of blank-term r

$low = 0, high = \max(B), r = \max(B)$
while $low < high$ **do**
  $mid = low + \frac{high - low}{2}$ # Fix blank-term
  $C_{EXP} = \sum_{i=0}^{|B|} \frac{b_i}{mid+1}$ # Required energy
  if $C_{EXP} > E_{EXP}$ **then** # Energy exceed E_{EXP}
    $low = mid + 1$
  else
    $high = mid$
    $r = \min(r, mid)$
  end if
end while
return r



(a) (b) (c)

Figure 6. User interface of SmartDC. The application displays a user's meaningful places on (a) the Google map, and (b) the list. It also visualizes (c) place information, including place name, connectivity, residence-time, and mobility history. The application called LifeMap is available in Android Market.

4 Evaluation

We evaluate SmartDC in three aspects: mobility learning, mobility prediction, and adaptive duty cycling. We mainly focus on the performance of prediction-based adaptive duty cycling, which is our major objective.

4.1 Implementation

In order to evaluate the proposed scheme, we implemented SmartDC on the Android SDK 2.1 running on commercial mobile phones equipped with GPS, GSM/CDMA, and Wi-Fi. Our implementation specifically focused on tracing a user's mobility based on intuitive UI design. Figure 6 shows a screen capture of the UI running on an HTC Desire. The system indicates a user's meaningful places on the Google map and the list to confirm/modify the place signatures, such as connectivity, residence time, place name, and mobility history.

In our experiment, we omitted the analysis of some of the parameters for the learning scheme, since our mobility learner uses a scheme similar to that of previous studies. The time interval t_w is set to two minutes, as suggested by [19]. The Wi-Fi scanning intervals and window size are 10 seconds and 30 seconds respectively, and the similarity threshold φ of the Wi-Fi vector is set to 0.7 [12]. The accuracy threshold ϕ for the WPS is set to 500 meters [17]. The GPS is activated for 30 seconds for single positioning, which is common in GPS usage.

To reduce computation overheads in adaptive duty cycling, we (1) converted the float value to an integer value with 10^{-3} precision, (2) used discrete time intervals in minutes, and (3) scaled down the energy budget, dividing it by the energy consumption of Wi-Fi scanning, which is the minimum cost in our scheme.

We set the energy budget E and the sensing cost c as follows: The maximum energy budget is (1,400 mA×3.7 V-34.5 mW)×3,600 s=18.5 kJ, which is the available battery capacity excluding the energy cost of idle state. If the battery level and energy constraints are 30% and 10%, the allowed energy budget is 18.5 kJ×0.3×0.1=555 J. Based on the energy profile in Section 2.1, the level 2 sensing cost is the energy consumption of Wi-Fi scanning: 114.5 mW×30 s=3.5 J. The level 3 sensing consumes 3.5 J+440.8 mW×30 s=16.7 J: the energy of level-2 sensing and reading GPS for 30 seconds.

4.2 Data Collection

We collected real traces from 8 graduate students over four weeks using HTC Hero, HTC Desire, and Samsung Galaxy S smartphones. SmartDC was running as a background service to automatically collect the user's mobility and to trace sensor usage time. Participants installed SmartDC on their primary phones. To collect the ground truth, the participants explicitly labeled the place names and kept a diary of places they had visited with the entrance and departure times. In the collected data traces, there are 651

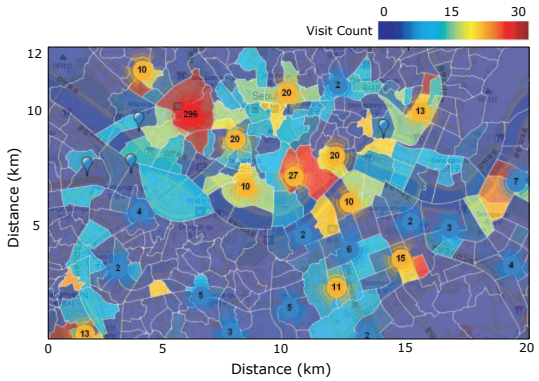


Figure 7. Collected data traces from 8 users in Seoul, Korea. The visit count is represented as color on each administrative section. The frequently visited sections include the university and homes of participants. The number in a circle represents the number of generated meaningful places in the near region.

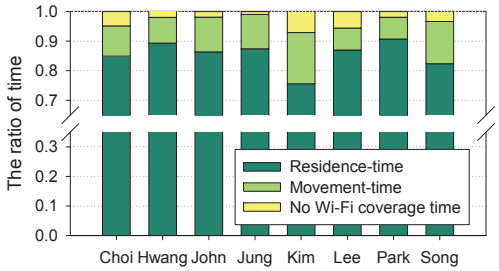


Figure 8. The ratio of residence-time, movement-time and no Wi-Fi coverage time. Participants spend approximately 85% of their time for staying. The ineffectivity rate of the proposed scheme is about 3.5%.

meaningful places, 9681 nodes in 1717 paths, and 52510 Wi-Fi APs. Figure 7 shows the collected data traces in Korea.

We measured the entropy of collected data traces. The proposed system is ineffective when a user is located in regions that do not have Wi-Fi coverage since the key scheme of our movement detector uses Wi-Fi scanning. Thus, we defined *ineffectivity rate* as the ratio of time spent by a user in regions without Wi-Fi coverage to the total spent time. The collected traces show that the ineffectivity rate is trivial, as illustrated in Figure 8. Participants spend about 3.5% of their time in non-Wi-Fi regions. Otherwise, 85% of their spent time is for staying behavior. This means that the system could reduce about 85% of energy consumption for location sensing if it could accurately predict the residence time at each place.

4.3 Mobility Learning

We evaluate the performance of mobility learning in two ways: place detection and energy consumption. To quantitatively evaluate the performance of detecting meaningful places, we use the following methodologies, which were used in [11]: *Perceived* places are recorded by a user in a diary for ground truth; *Discovered* places are generated by SmartDC; *Correct* places are recorded and discovered; *Merged* places belong to correct places, but two different places are discovered as a single place; *Divided* places belong to correct places, but a single place is discovered as several different places; *Missed* places are recorded but are not discovered places; *False* places are meaningless places, discovered but not recorded; *Interesting* places are meaningful places, discovered but not recorded. We further defined *accuracy* and *precision* as follows:

$$\text{accuracy} = \frac{|correct|+|interesting|}{|discovered|}, \text{precision} = \frac{|correct|-|merged|}{|perceived|}.$$

In words, accuracy is a measure of meaningful places correctly discovered by the system, and precision is a measure of discovered meaningful places, excluding merged places among the recorded places by a user.

Table 2. The Results of Place Detection

Metric	*Number of Places*	*Number of Visits*	*Amount of Residence time*
Correct	64	361	993.7h
Interesting	26	62	9.0h
Merged	3	4	1.3h
Divided	3	7	0.8h
Missed	14	23	11.1h
False	28	29	2.9h
Accuracy	0.76	0.93	0.99
Precision	0.78	0.93	0.99

The performance of place detection using a trace of three participants is shown in Table 2. The cases of interesting places are short-visit places, such as bus stations, subway stations, and toilets. The false places are normally generated in transit, due to traffic jams or non-static APs (e.g., Wi-Fi gateways in vehicles or locally hosted mobile devices). Since we use Wi-Fi vectors to detect place, missed places are located in regions that do not have Wi-Fi coverage, such as a restaurant in subways or suburban areas. Merged places are noticed in dense places near APs with a strong signal, such as adjacent rooms. The major factors of divided places are intermittent beacons in sparse AP deployments or changes in Wi-Fi infrastructures. As for the number of places, the accuracy and precision are 0.76 and 0.78, respectively. These increase to 0.93 and 0.99 when we consider the number of visits and the amount of residence time, respectively. The result indicates that collected data is sufficient to use as a mobility history for prediction. In addition, 80% of the visits are detected within a 139-second delay in entering, and a 161-second delay in departure, as illustrated in Figure 9(a). Figure 9(b) indicates that the error in mobility learning is represented as an approximated Gaussian distribution. We applied this distribution as a noise to mobility predictor to cover the error in mobility learning.

We present each sensor's energy consumption for mobility learning. Figure 10(a) shows the sensors' activate times in a day. The usage time of GSM is negligible (i.e., less than 20 seconds), because the computation time to update an LAC takes only a few milliseconds. GPS is typically activated in transit, since we turn on the GPS only if the system cannot obtain accurate location information from both the WPS and historical mobility. The active time of Wi-Fi is uniformly distributed due to the fixed time interval. The average usage times of Wi-Fi and GPS are 300 ± 12 minutes and 16 ± 7 minutes, as illustrated in Figure 10(b). The GPS usage time is trivial because WPS provides accurate location information in Seoul, Korea. Such usage time would be increased

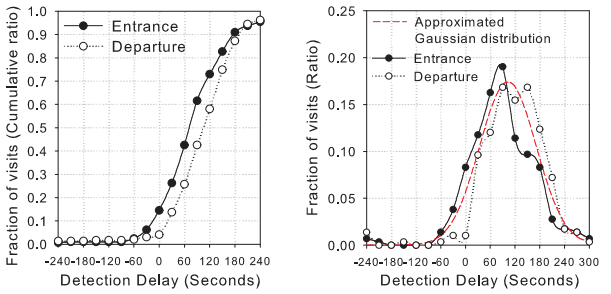


Figure 9. Detection delay of departure time and entrance time in (a) cumulative probability, and (b) probability distribution.

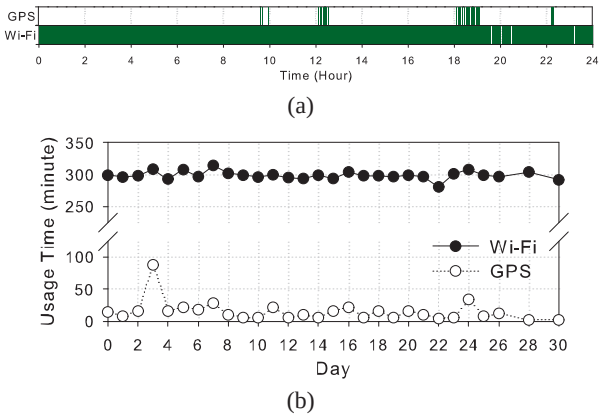


Figure 10. The (a) one day, and (b) daily usage pattern of sensors in mobility learner.

if a user lives in regions that do not provide sufficient WPS data. Although WPS has poor coverage, the GPS usage time would be gradually decreased because the proposed system reuses location information in historical data by matching Wi-Fi fingerprints of places. The average energy consumption is 1,831±228 J, which would reduce battery lifetime by 12±4% in real deployments. In the following section, we present the potential predictability of human mobility, and the effectiveness of prediction-based adaptive duty cycling to minimize such energy consumption.

4.4 Potential Predictability

We investigate the upper bound of predictability in individual mobility patterns to estimate the optimal performance of mobility prediction. To measure maximal probability, we consider that the current pattern is predictable if it is a previously observed one, since we use a history-based predictor. We defined two metrics: *revisit ratio R*, and *predictable movement ratio* R_m. R indicates the maximal predictability of location prediction, which is defined as:

$$R = \frac{No.\ of\ revisits}{No.\ of\ visits}.$$

Since our system determines the efficient sensing schedule, predictable residence time at a revisited place is a more important factor than the next location. Thus, we further measured each user's predictable movement ratio defined as:

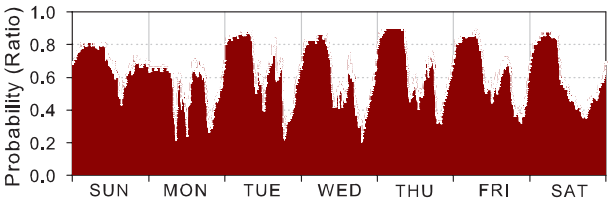


Figure 12. The probability of most visited location during the corresponding minute-long period. 0.84 at Tuesday 1am means that participants are located 84% of their time in his/her most visited location at Tuesday 1 am.

$$R_m = \frac{Sum.\ of\ previously\ observed\ residence\ time\ at\ each\ revisit}{Sum.\ of\ residence\ time\ at\ each\ visit}.$$

R_m indicates the maximal predictability of mobility prediction. We consider current residence-time to be predictable if the difference to historical data is within a ±3 deviation (68-second) of error in mobility learner.

Figure 11 represents the fundamental limit for each user's predictability. Higher probability means that higher predictability is inherent in the individual mobility pattern. The result indicates that residence time prediction R_m is less predictable than location prediction R. In other words, humans tend to revisit frequently visited places, but their staying behavior contains relatively more randomness. To understand potential regularity with temporal features, we segmented each week into 7 days×24 hours×60 minutes=10,080 minute-intervals. We measured the number of visits at the most visited locations in every minute. Figure 12 shows the average predictability of all participants. During night time (0–7 am), high predictability is shown as 81±8%, while the predictability of mealtime (12–1 pm and 6–7 pm) shows low predictability as 47±14%. The result reveals the behavior tendency of human; that is, human tends to return home at evening and spend night time at home. This predictability may be biased since our participants are all graduate students whose life patterns are rather irregular. Our future work will examine large scale data analysis with various jobs and age groupings.

In Figure 13, we show the daily predictability of all participants in CDF curves. In predictability about movement behavior R_m, 50% of days contain more than 60% predictability. This indicates that the system could, at most, reduce 60% of energy for location sensing if the predictor algorithm is 100% accurate. R_m increases if we exclude the result of the first two weeks of data. The predictability, however, did not change significantly, eventhough we considered only weekday mobility.

In the remainder of this paper, we omit the analysis on R. Instead, we focus on R_m since our goal is to determine a sensing schedule that is strictly connected to prediction about residence time.

4.5 Mobility Prediction

We evaluated the performance of mobility predictors using three metrics: *accuracy*, *energy reduction*, and *efficiency*. We set the energy budget to infinite to eliminate the effect of energy constraints. The predictor generates three outcomes: (1) The predicted schedule correctly senses the change of location within the error of mobility learning (*correct pre-*

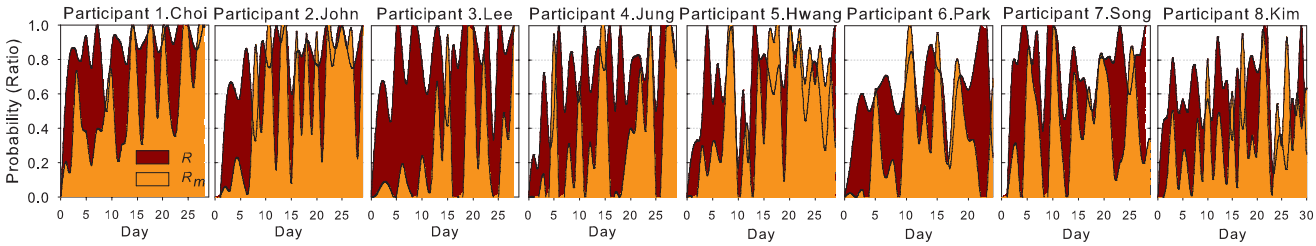
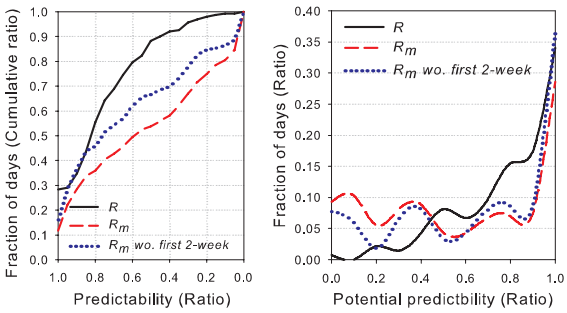


Figure 11. Potential predictability in each user's mobility pattern. Higher probability means that a user follows previously observed mobility patterns. Prediction about location R contains more predictable regularity than prediction about movement behavior R_m.



(a) (b)

Figure 13. The distribution of R and R_m in (a) CDF and (b) PDF. The leftmost curve in CDF indicates higher predictable regularity. The R_m is increased if we consider mobility without first two-week data.

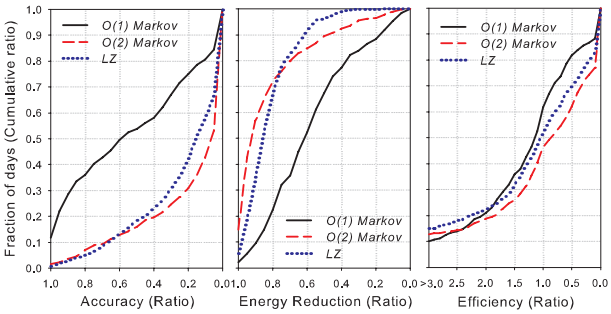


Figure 14. The performance of $O(1)$ Markov, $O(2)$ Markov, and LZ predictor under the infinite energy budget. The leftmost curves indicate better performance in accuracy, energy reduction and efficiency.

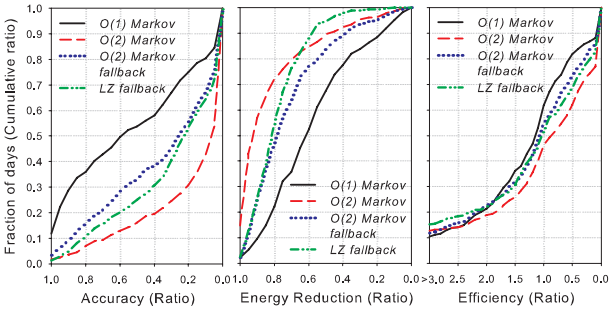


Figure 15. The performance of Markov and LZ predictors with fallback under the infinite energy budget.

diction); (2) The predicted schedule misses the moment of location change beyond the error range of mobility learning (*incorrect prediction*); (3) The predicted schedule is none, since the current pattern has not been seen before (*none-prediction*). The accuracy is defined as: $\frac{(1)}{(1)+(2)+(3)}$, and the energy reduction is defined as:

$$\text{energy reduction} = 1 - \frac{\textit{No. of sensing with predicted schedule}}{\textit{No. of sensing with fixed interval}}.$$

Since we assume that energy budget is infinite, accuracy indicates the maximal accuracy of predictors, and the energy reduction indicates the number of observed patterns using each predictor. Energy reduction converges to zero as the number of observed pattern increases. Finally, efficiency is defined as:

$$\text{efficiency} = \frac{\textit{accuracy}}{1 - \textit{energy reduction}}.$$

The optimal way derives high efficiency, which means the predictor precisely senses a user's mobility with a minimum number of sensings. We implemented and compared three location predictors: $O(1)$ Markov, $O(2)$ Markov, and LZ.

Under the condition of an infinite energy budget, $O(1)$ Markov predictor uses all observed patterns at current location, while both $O(2)$ Markov and LZ predictors use a sequential pattern of visited locations. We found that, counterintuitively, using more context (i.e., sequential patterns) is not efficient in predictions about residence time at current place, as illustrated in Figure 14. Although $O(2)$ Markov and LZ predictors used less energy than an $O(1)$ Markov predictor, they failed to predict many cases, and their efficiency is also not significantly outperformed. The result indicates that context about previously visited locations (e.g., workplace or restaurant) does not help to predict residence time at current places (e.g., home).

The none-prediction case also causes performance deterioration of $O(2)$ Markov that contains 13% none-prediction cases. To cover the effect of none-prediction, we applied a fallback mechanism, which uses the result of the low-order predictor if the high-order predictor has no prediction [25]. $O(2)$ Markov uses $O(1)$ for none-prediction, and LZ predictor uses fallback at leaf (i.e., suffix of current patterns). Indeed, fallback improved the accuracy and efficiency of $O(2)$ Markov as illustrated in Figure 15. LZ predictor slightly benefits from fallback because it only contains 3% none-prediction cases. The overall efficiency of $O(2)$ Markov with fallback, however, does not outperform $O(1)$.

We applied temporal state to predictors to investigate the correlation between human movement behavior and temporal features. We quantized time in 24 (24-hour) or 168 (24-

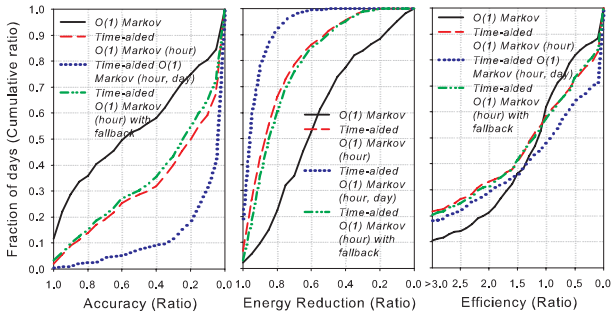


Figure 16. The performance of time-aided Markov under the infinite energy budget.

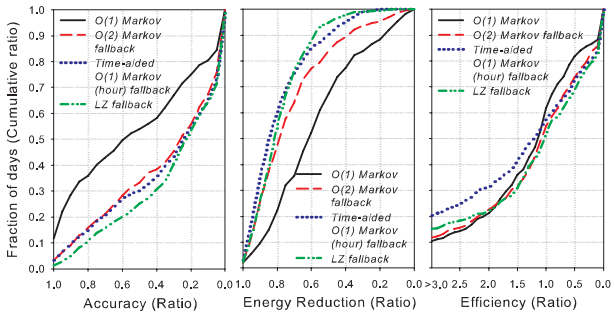


Figure 17. The performance of best variants of predictors under the infinite energy budget.

hour$\times$7-day) buckets. Then, the state of predictor forms location and time pairs. The fallback mechanism reduces a two-dimensional state (location, time) to one dimension (location) if time-aided Markov fails to predict. The assumption is that humans tend to move with daily or weekly regularity. Figure 16 shows the performance of time-aided Markov predictors. The accuracy of $O(1)$ Markov with 168-buckets was far from $O(1)$ due to the insufficient number of observed patterns (i.e., 15% none-prediction cases). Despite the smaller number of sensings, time-aided predictor's efficiency was clearly higher than $O(1)$. It meant that a time-aided predictor accurately filtered the redundant patterns for prediction about residence time. The fallback slightly increases the accuracy of a time-aided Markov predictor.

Figure 17 shows the performance of the best variants of predictors. The accuracy of $O(1)$ Markov significantly outperforms other predictors because of the numerous samples. The time-aided Markov shows high efficiency since it accurately filtered out redundant information using temporal features. Figure 18 indicates the cost of predictors. We ignored the database size in smartphones since it is the same for all predictors. We only considered observed pattern size, such as the observed k-length location sequence for Markov and the number of nodes for the LZ predictor. Considering the efficiency and cost of predictors, we chose $O(1)$ Markov as the simplest predictor and time-aided $O(1)$ Markov with 24-buckets as the high efficiency predictor. In the following, we evaluate prediction-based adaptive duty cycling with energy constraints.

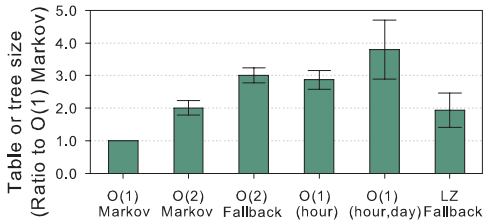


Figure 18. The predictor's memory usage. We present memory usage as the relative ratio to the number of entries for O(1) Markov (simplest method), since its absolute size is dependent on personal mobility pattern.

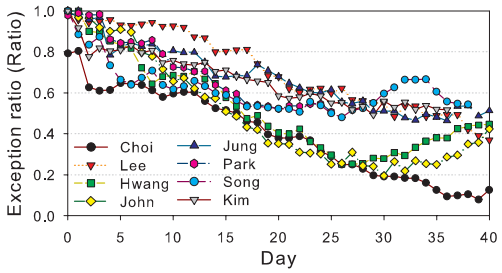


Figure 19. The exception ratio of each participant. The exception rate is gradually decreased as the mobility learning is continuous. The degree of declination is dependent on individual mobility pattern.

4.6 Adaptive Duty Cycling

In this section, we investigate the cost and performance of predication-based adaptive duty cycling. We analyze the sensitivity of parameters and the effectiveness of the prediction method based on user traces. We then show the energy consumption and accuracy of the overall system.

The proposed system splits the energy budget into two parts: prediction and exception. The system automatically derives a personalized exception ratio using the average of daily predictability R_m during two recent weeks. Figure 19 illustrates the measured exception ratio for each participant. The exception rate gradually decreases as the mobility data increases. Since the proposed system automatically learns a user's mobility, the exception ratio will be adaptively changed if a user changes his/her mobility pattern (e.g., change jobs, or summer vacation).

We ran the emulation to evaluate the effectiveness of the adaptive duty cycling with the following assumptions: (1) People use their smartphone for 28 hours with a single battery charge, which is the average lifetime of typical users' smartphones [8]; and (2) the system correctly recognizes revisited places. Figure 20 describes two steps in the emulation and three possible cases:

Case *PRD*: The predicted sensing schedule contains the ground truth of a location change moment. Detection delay is calculated using the first sensing moment after the ground truth.

Case *PRD-LRN*: The system uses a fixed interval after the predicted sensing schedule is finished.

Case *LRN*: The system uses a fixed interval since a place does not have mobility history.

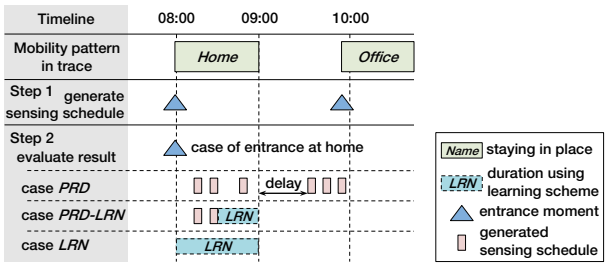


Figure 20. Two steps and possible outcomes in the emulation.

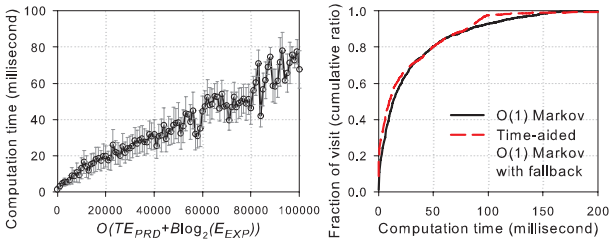


Figure 21. The computation time for processing prediction-based adaptive duty cycling.

We use the following notations to evaluate the performance of the prediction-based adaptive duty cycling:

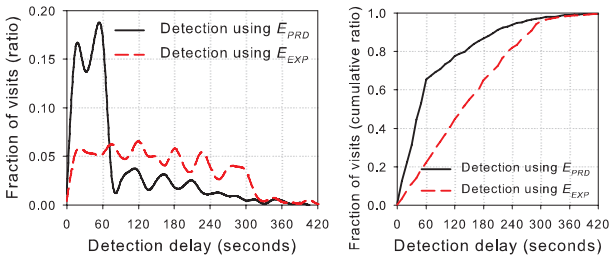
$\mathcal{N}$: Number of sensing events using Wi-Fi (cost).

$\mathcal{D}$: Place detection delay in seconds (accuracy).

We do not explicitly state the number of GPS readings since it is dependent on the location accuracy obtained in sensing level 2.

We first present the computation time of core functions in our algorithm. The computational complexity of adaptive duty cycling is $O(TE_{PRD} + B\log_2(E_{EXP}))$, where T is the maximal time among observed residence time, E_{PRD} is the energy budget for the prediction, E_{EXP} is the energy budget for the exception, and B is the number of blank-terms. The complexity is equal to the cost of filling a $T \times E_{PRD}$ array for dynamic programming and the cost of finding a minimum blank-term using a binary search. Here, the complexity of finding entries of a Markov scheme is $O(1)$ since we use a hash-table for it. Figure 21 presents the computation time on a Samsung Galaxy. The computation time is less than 100 milliseconds in 92% visits using $O(1)$ Markov, and 98% visits using time-aided $O(1)$ Markov with fallback. Such computation time is sufficient to process in real-time, without impeding users.

The energy budget constraint E is a major factor in determining an optimal sensing schedule. SmartDC limits the energy usage to less than the given energy budget. Thus, a smaller E may derive more missed places due to the limited sensing opportunity. The efficient usage of energy involves maximizing the number of detected location changes within smaller sensing delays. Figure 22 shows the emulation results according to various energy budgets. We considered the cases of *PRD* and *PRD-LRN*, but not the *LRN* case, since the cost and performance of the *LRN* case is equal to the learning scheme. Intuitively, a larger energy budget derives smaller detection delays and more sensing events. The specific performance is dependent on personal mobility pattern.



(a) (b)

Figure 23. The delay and detected type of all participants in (a) PDF and (b) CDF. The detection delay derived from E_{PRD} is significantly shorter than the delay derived from E_{EXP}.

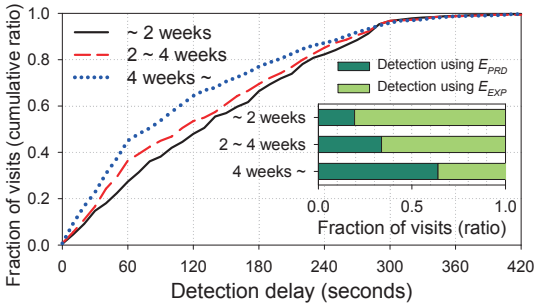


Figure 24. The detection delay corresponding to collection period. The inset figure indicates the ratio of detected type.

In the case of a 10% energy constraint, adaptive duty cycling consumes only 24±7% energy of the case using fixed interval, and still detects 80% of location changes within an additional 93±22 seconds. In addition, time-aided $O(1)$ Markov consumes 61±7% energy of the $O(1)$ Markov, and delay of 80% of the location change is 113±23 seconds. The result reveals that using temporal features is a more efficient way than using all historical data for predictions.

The major reason for energy saving is that (1) people tend to follow the observed pattern in mobility history, and, (2) our system successfully allocates a given energy budget based on individual mobility pattern. Figure 23 shows the correlation between detection delay and detected type (i.e., detection using E_{PRD} or E_{EXP}). The detection using E_{PRD} derives a smaller delay (i.e., the leftmost curve in Figure 23(b)). In other words, the system detects a user's movements within a small delay when a user follows his/her previous pattern. The result implicitly reveals that the system also covers the randomness of a user's movements by using E_{EXP}. The proposed system gradually decreases detection delay as the collection period increases, as illustrated in Figure 24. Indeed, the ratio of detection using E_{PRD} increases as the collection period increases.

Finally, we evaluated the performance of the overall system including cases *PRD*, *PRD-LRN*, and *LRN*. Figure 25 presents the energy usage ratio and detection delay. Compared to fixed time interval, the proposed system consumes 45±6% energy: 23±8% of energy for adaptive duty cycling and 22±8% of energy for learning. The system detects 80% and 90% of location change shows less than 160±20 and

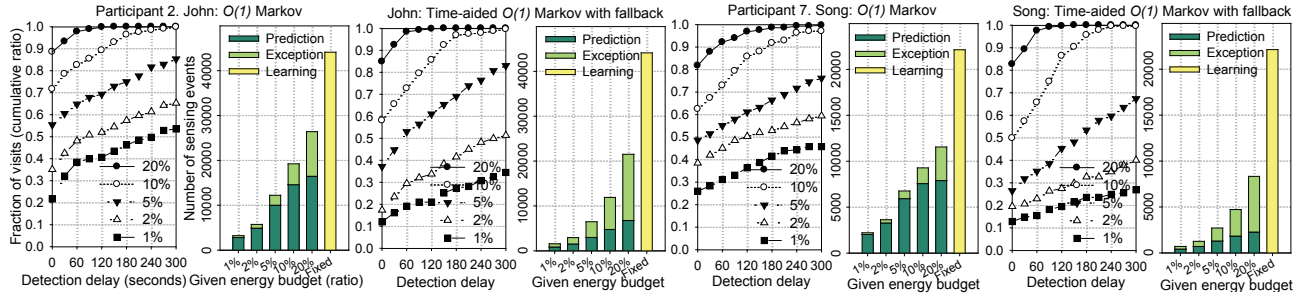
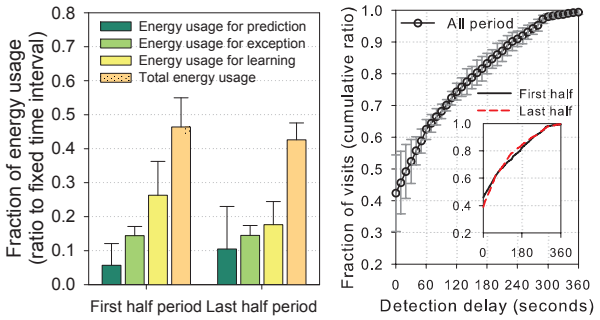


Figure 22. The cost and performance according to various energy budgets. The detection delay is additional delay compared to learning scheme. Zero second delay includes the cases that show less delay than for a fixed interval of learning scheme (i.e., 2-minute).



(a) (b)

Figure 25. (a) The energy usage ratio to fixed time interval and (b) detection delay. Zero second delay includes the *LRN* cases.

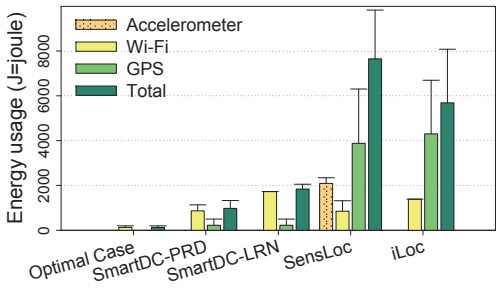


Figure 27. Daily energy usage. To estimate the energy usage of related systems, we utilized our energy profile and sensing policy in [12, 19]. The detailed sensing policy is presented in Table 1. The optimal case uses a manual sensing by a user.

226±24 seconds delay, respectively. Figure 25(a) indicates that the system reduces energy consumption as the collection period increases. Despite less energy consumption, the detection delay in the last-half period is similar to delay in the first-half period, as illustrated in Figure 25(b). It reveals that our system successfully detected a user's mobility using regular patterns without additional delay. Figure 26 shows the sensors' activate time of a day among collected data. Compared to the results from learning schemes, as shown in Figure 10(a), the sensor active time is reduced based on the predicted time of a location change. The sensing schedule during night-time specifically reveals that the proposed system successfully predicts residence-time at the revisited places: it allocates E_{PRD} around the office-going hour while E_{EXP} covers the nighttime (10pm to 6am). Note that determining the dynamic time interval in a learning scheme in accordance with battery levels will be our challenge in future works. The average usage time of Wi-Fi is 180±55 minutes. The average energy consumption for one day is 974±352 J, which is 47% less energy consumption than learning schemes without prediction.

Figure 27 shows the comparison of energy usage among the related systems. In sensor usage aspects, the proposed system consumes 81% less energy than the periodic sensing schemes without accelerometer [19], and 87% less energy than a scheme that employs context-aware sensing [12], while additional detection delay is about 160 seconds. The major factors for energy saving are (1) adaptive duty cycling based on mobility prediction, and (2) multiple sensing levels to reuse location information without activating power-intensive sensors. Without energy consumption on GPS for path tracking, our approach consumes 38% less energy than the periodic sensing schemes, and 70% less energy than context-aware schemes. Considering the optimal case (a user manually turns on sensors when he/she changes location), we still have room for improvement, as illustrated in Figure 27. In future work, we plan to design a mobility model examining behavior tendencies of human to approach the optimal case.

In summary, our approach successfully saves energy for everyday location monitoring, and detects 80% of location changes within a 140- to 180-second delay, depending on given energy constraints and individual mobility patterns. The effectiveness of such delays depends on the application. This delay is viable in applications that use human mobility patterns, such as social applications, healthcare applications, environment-related applications, epidemics, and urban planning. Applications requiring sensitive delays, such as reminder applications and advertisement services, are minimally affected by such delays. While the proposed system uses shorter time intervals for reducing detection delays, the system still contributes to minimizing energy consumption through prediction-based adaptive duty cycling.

5 Conclusion

In this paper, we proposed SmartDC to solve the energy issue of continuous sensing in real deployments. To the best of our knowledge, we are the first to implement a practical system that simultaneously performs mobility learning and

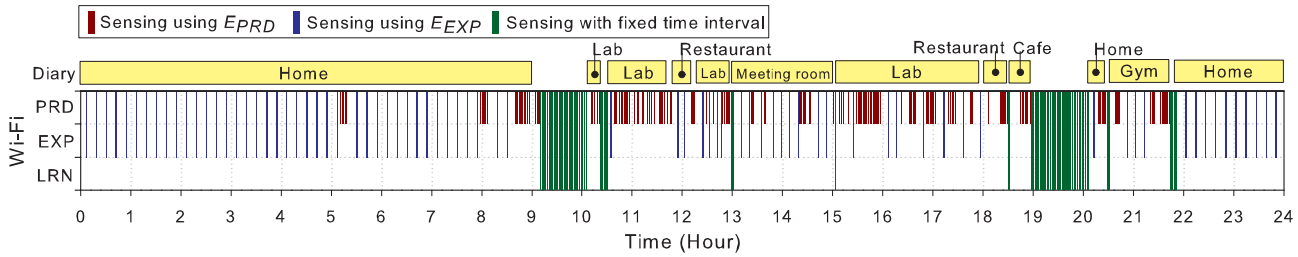


Figure 26. Sensor active time of prediction-based adaptive duty cycling. The diary indicates the ground truth of a user's mobility.

prediction for everyday location monitoring, based on off-the-shelf smartphones. We designed sensing levels and an adaptive duty cycle using MDP with automatically collected mobility data. The experiments show that our approach saves 81% to 87% energy over previous work, while its place detection delay is increased by approximately 160 seconds. We believe that our approach can be used as a building block toward expanding the domain of mobile services and to gathering individual human mobility patterns for research purposes. In addition, our prediction-based approach is suitable for a wide range of emerging mobile applications related to the temporal predictability. Our future work includes the design of advanced mobility model to improve the performance of mobility prediction.

6 Acknowledgments

This work was supported by Microsoft Research Asia and the National Research Foundation of Korea grant funded by the Korean government, Ministry of Education, Science and Technology (No.2011-0000156, No.2011-0015332).

7 References

[1] F. Alizadeh-Shabdiz and E. J. Morgan. System and method for estimating positioning error within a wlan-based positioning system, Patent US 2008/0 108 371 A1, 2008.

[2] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw.*, MobiCom'09, pages 261–272. ACM, 2009.

[3] A. Bhattacharya and S. Das. Lezi-update: an information-theoretic framework for personal mobility tracking in pcs networks. *ACM/Kluwer Wireless Networks*, 8:121–135, 2002.

[4] J. Biesterfeld, E. Ennigrou, and K. Jobmann. Neural networks for location prediction in mobile networks. In *Proc. Int. Workshop Appl. Neural Netw. Telec. 3*, pages 207–214. Lawrence Erlbaum, 1997.

[5] Y. Chon and H. Cha. Lifemap: A smartphone-based context provider for location-based services. *IEEE Pervas. Comput.*, 10(2):58–67, Apr-Jun 2011.

[6] Y. Chon, E. Talipov, and H. Cha. Autonomous management of everyday places for personalized location provider. *IEEE Trans. Syst., Man. Cyber. C, Appl. Rev.*, in print, 2011.

[7] I. Constandache et al. Enloc: Energy-efficient localization for mobile phones. In *INFOCOM'09*, pages 2716–2720. IEEE, 2009.

[8] H. Falaki et al. Diversity in smartphone usage. In *Proc. 7th Int. Conf. Mobile Syst., Appl., Serv.*, MobiSys'09, pages 179–194. ACM, 2010.

[9] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[10] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[11] D. H. Kim, J. Hightower, R. Govindan, and D. Estrin. Discovering semantically meaningful places from pervasive rf-beacons. In *Proc. 11th Int. Conf. Ubiquit. Comput.*, UbiComp'09, pages 21–30. ACM, 2009.

[12] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *Proc. 8th ACM SenSys'10*, pages 43–56. ACM, 2010.

[13] N. Klepeis et al. The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. *J. Expo. Anal. Environ. Epidemiol.*, 11(3):231–252, May-Jun 2001.

[14] A. LaMarca et al. Place lab: Device positioning using radio beacons in the wild. In *Lecture Notes in Computer Science*, volume 3468, pages 116–133, Munich, 2005.

[15] N. Lane et al. A survey of mobile phone sensing. *IEEE Commun.*, 48(9):140 –150, Sept. 2010.

[16] J.-K. Lee and J. C. Hou. Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application. In *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, MobiHoc'06, pages 85–96. ACM, 2006.

[17] K. Lin et al. Energy-accuracy trade-off for continuous mobile device location. In *Proc. 8th Int. Conf. Mobile Syst., Appl., Serv.*, MobiSys'10, pages 285–298. ACM, 2010.

[18] H. Lu et al. The jigsaw continuous sensing engine for mobile phone applications. In *Proc. 8th ACM Conf. Embedded netw. Sens. Syst.*, SenSys'10, pages 71–84. ACM, 2010.

[19] Y. Ma, R. Hankins, and D. Racz. iloc: a framework for incremental location-state acquisition and prediction based on mobile sensors. In *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, CIKM'09, pages 1367–1376. ACM, 2009.

[20] A. Nicholson and B. Noble. Breadcrumbs: forecasting mobile connectivity. In *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, MobiCom'08, pages 46–57. ACM, 2008.

[21] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proc. 8th MobiSys'10*, pages 299–314. ACM, 2010.

[22] B. Priyantha, D. Lymberopoulos, and J. Liu. LittleRock: Enabling Energy-Efficient Continuous Sensing on Mobile Phones. *IEEE Pervas. Comput.*, 10(2):12–15, Apr-Jun 2011.

[23] A. Rahmati and L. Zhong. Context-based network estimation for energy-efficient ubiquitous wireless connectivity. *IEEE Trans. Mobile Comput.*, 10(1):54 –66, 2011.

[24] C. Song et al. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[25] L. Song, D. Kotz, R. Jain, and X. He. Evaluating next-cell predictors with extensive wi-fi mobility data. *IEEE Trans. Mobile Comput.*, 5(12):1633 –1649, Dec. 2006.

[26] P. Wang, M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding the spreading patterns of mobile phone viruses. *Science*, 324(5930):1071–1076, 2009.

[27] Q. Yuan, I. Cardei, and J. Wu. Predict and relay: an efficient routing in disruption-tolerant networks. In *Proc. 10th ACM MobiHoc'09*, pages 95–104. ACM, 2009.

[28] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving energy efficiency of location sensing on smartphones. In *Proc. 8th MobiSys'10*, pages 315–330. ACM, 2010.

[29] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans. Inf. Theory*, 24(5):530 – 536, Sep 1978.